AENSI Journals

Australian Journal of Basic and Applied Sciences

ISSN:1991-8178

Journal home page: www.ajbasweb.com

Algorithm Simulation Supported on Demodulation Border and Hough Commute for a Rectangular Situation

Mohammad Kiani and Mehdi Kiani

Department of Mechanical Engineering, Jawaharl Nehru Technologial University Hydrabad, India.

ARTICLE INFO

Article history:
Received 25 June 2014
Received in revised form
8 July 2014
Accepted 25 July 2014
Available online 20 August 2014

Keywords:
Edge detecting, Hough transform, rectangle positioning, simulation.

ABSTRACT

Edge detection is the name for a set of mathematical methods which aim at identifying points in a digital image at which the image brightness changes sharply or, more formally, has discontinuities. In this paper, a surprisingly simple yet powerful procedure, based on the Standard Hough transmute, is proposed to path oblong objects. It works on the rim image obtained by applying the Canny rim tracer to the source image. The oblong object in the rim image is then extracted firmly-fixed on the Standard Hough Transform and related conditions of a oblong. Finally, the area of interest is shifted to the new position where the tracked object is located. This paper is organized as follows. The next department details our proposed procedure. The present accumulator has summits that may have values greater than the determined threshold. The simplest way to find these summits is to compare the values of the summits with the threshold value. Then the summits with the highest values are compared to other summits in the neighborhood using predefined windows.

To Cite This Article: Mohammad Kiani and Mahdi Kiani., Algorithm Simulation Supported on Demodulation Border and Hough Commute for a Rectangular Situation. ***Aust. J. Basic & Appl. Sci.,*** *8(13): 570-576, 2014*

INTRODUCTION

Object tracking is an important task within the field of computer vision. The proliferation of high-powered computers, the availability of high quality and inexpensive video cameras, and the increasing need for automated video analysis has generated a great deal of interest in object tracking algorithms. Edge detection is the name for a set of mathematical methods which aim at identifying points in a digital image at which the image brightness changes sharply or, more formally, has discontinuities (Baddeley, A., 1992; Yang, C., 2005).

The points at which image brightness changes sharply are typically organized into a set of curved line segments termed edges. The same problem of finding discontinuities in ID signals is known as step detection and the problem of finding signal discontinuities over time is known as change detection. Edge detection is a fundamental tool in image processing, machine vision and computer vision, particularly in the areas of feature detection and feature extraction (Haralick, R.M., 1983). Although certain literature has considered the detection of ideal step edges, the edges obtained from natural images are usually not at all ideal step edges. Instead they are normally affected by one or several of the following effects: focal blur caused by a finite depth-of-field and finite point spread function. Penumbral Blur caused by shadows created by light sources of non-zero radius. shading at a smooth object a number of researchers have used a Gaussian smoothed step edge (an error function) as the simplest extension of the ideal step edge model for modeling the effects of edge blur in practical applications. Thus, a one-dimensional image f which has exactly one edge placed at $x = 0$ may be modeled as:

$$f(x) = \frac{I_r - I_l}{2}\left(\operatorname{erf}\left(\frac{x}{\sqrt{2}\sigma}\right) + 1\right) + I_l.$$

At the left side of the edge, the intensity is $I_l = \lim_{x \to -\infty} f(x)$, and right of the edge it is $I_r = \lim_{x \to \infty} f(x)$. The scale parameter σ is called the blur scale of the edge.

This means that there is no inherent way to judge what the edges are in a given discrete image. Therefore, the concept of an image edge is only what we define it to be. Derivative approximations were computed either from the pixels directly, using operators such as Robert's cross operator (Roberts 1965), the Sobel operator (Pingle 1969) and the Prewitt operator (Prewitt 1970), or from local least-squares fitting (Haralick 1984). An important step was then taken by the introduction of multi-scale techniques. (Torre and Poggio 1980) motivated

Corresponding Author: Mohammad Kiani, Department of Mechanical Engineering, Jawaharl Nehru Technologial University Hydrabad, India.

Australian Journal of Basic and Applied Sciences, 8(13) August 2014, Pages: 570-576

the need for linear filtering as a pre-processing step to differentiation, in order to regularize ill-posed differentiation into well-posed operators (Lappas, P., 2001). The Marr-Hildreth operator (Marr and Hildreth 1980) was motivated by the rotational symmetry of the Laplacian operator and the special property of the Gaussian kernel as being the only real function that minimizes the product of the variances of the filter in the spatial and the Fourier domains.1 Other early techniques with a multi-scale character were presented by (Rosenfeld and Thurston 1971) and by (Marr 1976). (Canny 1986) considered the problem of determining an \optimal smoothing filter" of finite support for detecting step edges. The methodology was to maximize a certain performance measure constituting a trade-off between detection and localization properties given a restriction on the probability of obtaining multiple responses to a single edge. He also showed that the resulting smoothing filter could be well approximated by the first-order derivative of a Gaussian kernel. (Deriche 1987) extended this approach to filters with infinite support, and proposed a computationally efficient implementation using recursive filters.2 (Canny 1986) also introduced the notions of non-maximum suppression and hysteresis thresholding. Similar concepts were developed independently by (Korn 1988). A smaller amount of smoothing, on the other hand, can be expected to improve the localization properties at the cost of a lower signal-tonoise ratio (Haralick, R.M., 1984). To circumvent this problem, (Bergholm 1987) proposed to detect edges at coarse scales, and to follow these to finer scales using edge focusing. He did, however, not present any method for determining the scale level for the detection step, or to what scales the edges should be localized. Hence, edge focusing serves mainly as a selection procedure, which among all the edges at the finer (localization) scale selects those who can be traced to corresponding edges at the coarser (detection) scale. The subject of this article is to address the general problem of automatically selecting local appropriate scales for detecting edges in a given data set. As will be illustrated by examples later in section 4, the choice of scale levels can crucially affect the result of edge detection. Moreover, different scale levels will, in general, be needed in different parts of the image. Specifically, coarse scale levels are usually necessary to capture diffuse edges, due to shadows and other illumination phenomena.

Second-Order Approaches to Edge Detection:

A more refined second-order edge detection approach which automatically detects edges with sub-pixel accuracy, uses the following differential approach of detecting zero-crossings of the second-order directional derivative in the gradient direction:

Following the differential geometric way of expressing the requirement of non-maximum suppression proposed by Lindeberg, let us introduce at every image point a local coordinate system (u, v), with the v-direction parallel to the gradient direction. Assuming that the image has been pre-smoothed by Gaussian smoothing and a scale space representation $L(x, y; t)$ at scale t has been computed, we can require that the gradient magnitude of the scale space representation, which is equal to the first-order directional derivative in the v-direction L_v, should have its first order directional derivative in the v-direction equal to zero

$\partial_v(L_v) = 0$

While the second-order directional derivative in the v-direction of L_v should be negative, i.e.,

$\partial_{vv}(L_v) \leq 0.$

Written out as an explicit expression in terms of local partial derivatives L_x, L_y ... L_{yyy}, this edge definition can be expressed as the zero-crossing curves of the differential invariant

$L_v^2 L_{vv} = L_x^2 L_{xx} + 2 L_x L_y L_{xy} + L_y^2 L_{yy} = 0,$

that satisfy a sign-condition on the following differential invariant

$L_v^3 L_{vvv} = L_x^3 L_{xxx} + 3 L_x^2 L_y L_{xxy} + 3 L_x L_y^2 L_{xyy} + L_y^3 L_{yyy} \leq 0$

where L_x, L_y ... L_{yyy} denote partial derivatives computed from a scale space representation L obtained by smoothing the original image with a Gaussian kernel. In this way, the edges will be automatically obtained as continuous curves with sub-pixel accuracy. Hysteresis thresholding can also be applied to these differential and subpixel edge segments (Haralick, R.M., 1983).

In practice, first-order derivative approximations can be computed by central differences as described above, while second-order derivatives can be computed from the scale space representation L according to:

$L_{xx}(x, y) = L(x - 1, y) - 2L(x, y) + L(x + 1, y).$

$L_{xy}(x, y) = (L(x-1, y-1) - L(x-1, y+1) - L(x+1, y-1) + L(x+1, y+1))/4,$

$L_{yy}(x, y) = L(x, y - 1) - 2L(x, y) + L(x, y + 1).$

Corresponding to the following filter masks:

$$L_{xx} = \begin{bmatrix} 1 & -2 & 1 \end{bmatrix} * L \quad \text{and} \quad L_{xy} = \begin{bmatrix} -1/4 & 0 & 1/4 \\ 0 & 0 & 0 \\ 1/4 & 0 & -1/4 \end{bmatrix} * L \quad \text{and} \quad L_{yy} = \begin{bmatrix} 1 \\ -2 \\ 1 \end{bmatrix} * L.$$

Higher-order derivatives for the third-order sign condition can be obtained in an analogous fashion. In either tracking approach, the objects are represented using the shape and/or appearance models described in

Australian Journal of Basic and Applied Sciences, 8(13) August 2014, Pages: 570-576

Section 2. The model selected to represent object shape limits the type of motion or deformation it can undergo. For example, if an object is represented as a point, then only a translational model can be used. In the case where a geometric shape representation like an ellipse is used for the object, parametric motion models like affine or projective transformations are appropriate. These representations can approximate the motion of rigid objects in the scene. For a non rigid object, silhouette or contour is the most descriptive representation and both parametric and nonparametric models can be used to specify their motion.

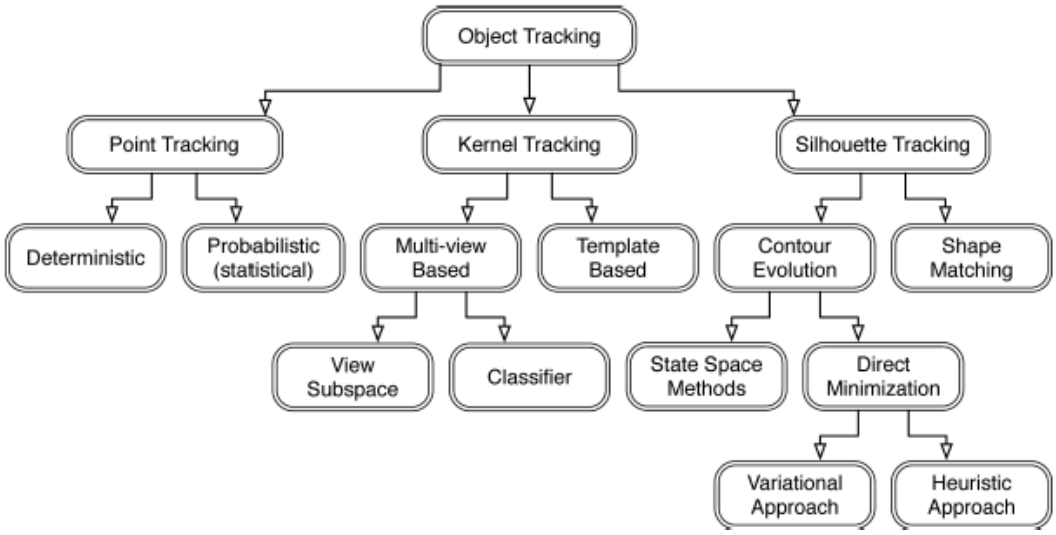

Fig. 1: Taxonomy of tracking methods.

Discussion of the Experimental Results:

In this paper, a surprisingly simple yet powerful procedure, based on the Standard Hough transmute, is proposed to path oblong objects. It works on the rim image obtained by applying the Canny rim tracer to the source image. The oblong object in the rim image is then extracted firmly-fixed on the Standard Hough Transform and related conditions of a oblong. Finally, the area of interest is shifted to the new position where the tracked object is located. This paper is organized as follows. The next department details our proposed procedure. Department III presents the discourse. Experimental results are in Section IV. This paper is drawn to a conclusion and aftertime work presented in Section V.Our procedure tracks an object firmly-fixed on its boundary. This type of tracking technique usually utilizes edges as the representative feature. A consequential characteristic of edges is that they are less susceptible to illumination changes, when compared to color features. Edge detection is the first footstep of our proposed procedure. The sapiential edge tracer is selected due to its simplicity and precision. Then, the search window is defined manually at the situation of the object to be tracked. We attempt to emblazon a window W outside the tracking object, or tracking oblong, O where the center of the window coincides with the center of the tracking oblong. The related equations are shown as follows:

In the next sections, the Standard Hough convert algorithm is applied to the search window W to detect direct lines, and the related conditions of an oblong help detect the oblong object in the corresponding area. Henceforth, the object to be path is determined and the step of defining the search window is revisited. The Hough convert has been the most beloved algorithm utilized to squeezed juice global features, such as direct lines, circles, and ellipses, from an image. This algorithm uses an array called parameter space to detect the existence of a script. For each pixel in the rim image, the Hough convert algorithm determines if there is enough document of a script at that pixel. It computes the parameter values and accumulates the cells in the parameter space according to all the pixels. Then, by finding the cells with the highest valences, the most likely script can be extracted.

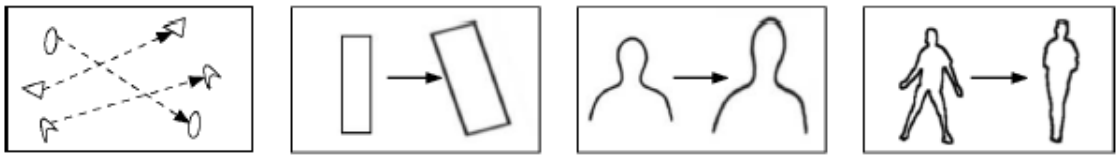

Fig. 2: Different tracking approaches.

Australian Journal of Basic and Applied Sciences, 8(13) August 2014, Pages: 570-576

The simplest way to find these peaks is to exert some form of threshold. Finally, the segments are extracted using the data obtained from the maximum situation. The basic Hough convert, also called the Standard Hough convert (SHT) has established itself as the default technique for straight script Hough convert assessment. Its amicability comes from its robustness to noise and the very simple algorithmic implementation of SHT. In SHT all the points (x; y) are calculated using the equation expressed in polar coordinates to obtain With a change in the substitution of in the parameter space.

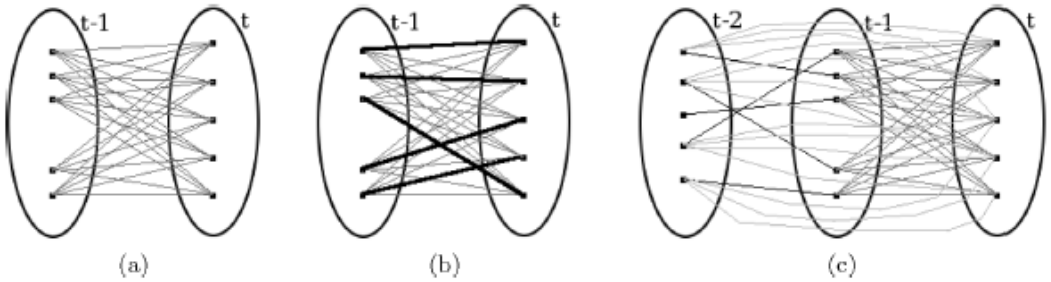

Fig. 3: Point correspondence. (a) All possible associations of a point (object) in frame t − 1 with points (objects) in frame t, (b) unique set of associations plotted with bold lines, (c) multiframe correspondences.

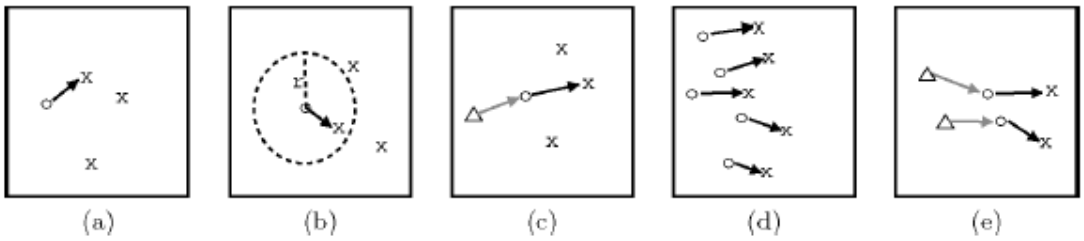

Fig. 4: Different motion constraints. (a) proximity, (b) maximum velocity (r denotes radius), (c) small velocity-change, (d) common motion, (e) rigidity constraints. denotes object position at frame t − 2, ◦ denotes object position at frame t − 1, and finally × denotes object position at frame t.

With the values (;) determined, the SHT algorithm then glances for the accumulator's cellules into which the parameters fall, and increments the values of those cellules. The conversion of an image I to a Hough space H can be represented as line have been accumulated, the repartition of the parameter space is shaped like a license. The most likely direct line in the image domain is deputize by the highest value in the parameter space. The peak lies at the center of the license. When all the aggregate points form a disordered parameter place, several overlapping butterflies are acquired.

The present accumulator has summits that may have values greater than the determined threshold. The simplest way to find these summits is to compare the values of the summits with the threshold value. Then the summits with the highest values are compared to other summits in the neighborhood using predefined windows. Straight lines are detected via these summits: For simplicity, we emblazon an internal circle Cinner = (xci; yci; rci) for the search window W. The internal circle is an attempt to eliminate redundant pixels that are inside the tracked object. This work reduces the calculation time of the Hough convert; it reduces noise that intervene with the precision of the tracking process. The center of this circle also coincides with the search window, An presumption is provided in this step: the number of noises in the search window is sufficient and thus several straight lines are detected. The list of script detected by SHT is sorted in descending instruction with respect to accumulating value. This work leads to a supposition that the most likely line has the biggest accumulating value. Consequently, we select the first line li in the list of detected lines, as a yardstick to eliminate disturbed lines. From this yardstick, we attempt to find the first line lj that is perpendicular to li in PL. The two li and lj are used to delete the placentas that contain lines equal to these two standard lines. Finishing, we have P0 L that contains the best candidates to form oblong. The first experience is run on a synthetic trail of 99 frames. The context containing pixel-noise in every frame is generated using a Gaussian accidental Number Generator (RNG). The mean, variance, and number of noise pixels of the RNG are 0, 300, and 7000, respectively.

The moving object is an oblong initially placed along the bottom-left to top-right oblique of the image. The oblong is 53 pixels in width, and 39 pixels in height. There is no noise inside this object. Fig. 4 shows four specimen frames from the trail of 99 frames. The first frame of the trail is shown in Fig. 4(a). The object moves with an approximate angle of 30 gradation to the left.

Australian Journal of Basic and Applied Sciences, 8(13) August 2014, Pages: 570-576

It shows the object before it palpate the right side. Then, the object moves to the left with an approximate angle of 45 gradation. It hits the top side at frame 51. Finally, the oblong object attempts motion back along the primary direction. In frame 94, the object is next to the left side. Our method is used to track this oblong object.

Corresponding parameters of the search window are (xz, yz) = (70, 160) and rw = 12 pixels. SHT, the parameters to detect lines are = 2 pixel, = 0.02 radian, and number of maximum detected lines linesmax =The result illustrate the proposed procedure perfectly paths the moving object throughout the trail of frames.

Four detected lines representing a oblong fit to four rim, successfully tracking the object. The result of four sample frames is shown in Fig. 3, where the border of the detected oblong is a red color. The frame rate of the resultant trail is almost 12 fps. In similar experiments, when we variation the number of noise pixels to the values that are greater than 7000, the tracking oblong will be missed. The second experience is run on a real trail of 62 frames.

These frames were take using a Logitech 962403-0211 QuickCam Fusion camera. The frame trail shows a man holding a box. The man motion the box from the center of the opinion, to the right and then returns to the prime situation. One side of the box is tracked. It has the shape of a oblong 97 pixels in width, and 127 pixels in height. There is no noise internal this oblong.

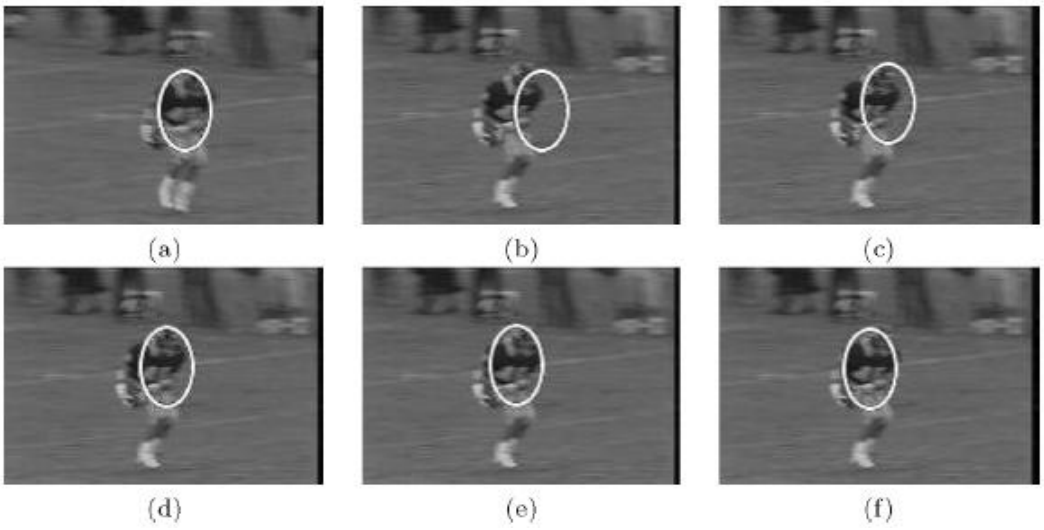

Fig. 5: Mean-shift tracking iterations. (a) estimated object location at time t − 1, (b) frame at time t with initial location estimate using the previous object position, (c), (d), (e) location update using mean-shift iterations, (f) final object position at time t.

The left column of Fig. 3 shows four sample frames from the trail of 62 frames. The first frame where the object was placed initially near the center of the view is shown in Fig. 3(a). Frames 40, 60 and 60 show the next situation of the object. The proposed procedure path the oblong that is one side of the box.

The parameters used in this experience are the same as in the prior one except (xz, yz) = (135, 127) and rw = 12 pixels. The result of four sample frames in the left pillar is shown in the right pillar of Fig. 4. The oblong that is tracked correctly is bordered in red. If the oblong is missed, a pink border is drawn at the prior situation. Fig. 5(d) shows that the object is missed and is bordered by a pink oblong.

43 of 65 (72%) tracked frames are detected. The frame rate of the resultant trail approximates 11 fps. Although the 72% detection rate is not very persuasive with chosen parameters, but we need to consider that noise exists in the real surroundings. The noise can come from illumination changes or background movement.

Conclusion:

When the object moves, it may meet a background that has a analogous severity or color. In this case, the rim detector might beget noise in the intersection area between the object and the background. We see in the right hand side, the rim of the tracked object and the temperature oblong are mistaken.

The third difficulty relates to noise inside the tracked object. We sketch this difficulty in Fig. 3. Fig. 4(a) shows an oblong with a word inside. This word is considered as noise. Two oblong are detected in this case. The red and solid border represents the correctly detected object, and the black and dashed border deputize the noise-oblong.

The center of this object will not be at the center of the correctly detected oblong. Fig. 3(c) and Fig. 5(d) show a blue circle as the center of the object. The measure of the search window is computed firmly-fixed on

Australian Journal of Basic and Applied Sciences, 8(13) August 2014, Pages: 570-576

the object's center. In this case, the window may miss one or more rim of the tracked oblong. We will lose the object to be tracked in the attendance of this type of noise.

REFERENCES

Arcelli, C. and G. Sanniti di Baja, 1992. Ridge points in Euclidean distance maps. Pattern Recog- nition Letters, 13(4): 237-242.

Baddeley, A., 1992. Errors in binary images and an l version of the haus- dorff metric. NieuwArchiefvoorWiskunde, 10: 157-183.

Haralick, R.M., 1983. Ridges and valleys in digital images. Computer Vision, Graphics, and Image Processing, 22: 28-38.

Haralick. R.M., 1984. Digital step edges from zero-crossings of second directional derivatives. IEEE Trans. Pattern Analysis and Machine Intell., 6.

Hills, M., T. Pridmore and S. Mills, 2003. "Object Tracking through a Hough space," Proceedings of International Conference on Visual Information Engineering (VIE), pp: 53-56.

Koenderink, J.J. and A.J. van Doorn, 1992. Generic neighborhood operators. IEEE Trans. Pattern Analysis and Machine Intell., 14(6): 597-605.

Lappas, P., J.N. Carter and R.I. Damper, 2001. "Object Tracking via the Dynamic Velocity Hough Transform," Proceedings of IEEE International Conference on Image Processing (ICIP), 2: 371-374.

Vapnik, V., 1998. Statistical Learning Theory. John Wiley NY.

Vaswani, N., A. Roychowdhury and R. Chellappa, 2003. Activity recognition using the dynamics of the configuration of interacting objects. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 633-640.

Yang, C., R. Duraiswami and L. Davis, 2005. "Fast Multiple Object Tracking via a Hierarchical Particle Filter," Proceedings of the 10th IEEE International Conference on Computer Vision (ICCV), 1: 212-219.

APPENDIX

```
functiondisplayTrackingResults()
% Convert the frame and the mask to uint8 RGB.
frame = im2uint8(frame);
mask = uint8(repmat(mask, [1, 1, 3])) .* 255;

minVisibleCount = 8;
if ~isempty(tracks)

% Noisy detections tend to result in short-lived tracks.
% Only display tracks that have been visible for more than
% a minimum number of frames.
reliableTrackInds = ...
                [tracks(:).totalVisibleCount] >minVisibleCount;
reliableTracks = tracks(reliableTrackInds);

% Display the objects. If an object has not been detected
% in this frame, display its predicted bounding box.
if ~isempty(reliableTracks)
% Get bounding boxes.
bboxes = cat(1, reliableTracks.bbox);

functionupdateUnassignedTracks()
fori = 1:length(unassignedTracks)
ind = unassignedTracks(i);
tracks(ind).age = tracks(ind).age + 1;
tracks(ind).consecutiveInvisibleCount = ...
tracks(ind).consecutiveInvisibleCount + 1;
end
end

% Get ids.
ids = int32([reliableTracks(:).id]);

% Create labels for objects indicating the ones for
```

Australian Journal of Basic and Applied Sciences, 8(13) August 2014, Pages: 570-576

```
% which we display the predicted rather than the actual
% location.
labels = cellstr(int2str(ids'));
predictedTrackInds = ...
                    [reliableTracks(:).consecutiveInvisibleCount] > 0;
isPredicted = cell(size(labels));
isPredicted(predictedTrackInds) = {' predicted'};
labels = strcat(labels, isPredicted);

% Draw the objects on the frame.
frame = insertObjectAnnotation(frame, 'rectangle', ...
bboxes, labels);

% Draw the objects on the mask.
mask = insertObjectAnnotation(mask, 'rectangle', ...
bboxes, labels);
end
end

% Display the mask and the frame.
obj.maskPlayer.step(mask);
obj.videoPlayer.step(frame);
end
```